I. Introduction and motivations

Goal: build an effective group of cryptographic strength, resisting all known attacks.

Dream: find Nechaev groups $G$, in which the best attack will be $O(\sqrt{q})$ (existence?)

Best groups so far: hyperelliptic curves of genus $g$, with size $\approx q^g$ over some finite field $\mathbb{F}_q$. Typical size $q^g \approx 2^{64} \sim 10^{19} \sim 6$.

- Miller, Koblitz (1986): elliptic curves are suggested for use, following the breakthrough of Lenstra in integer factorization (1985).

In this series of talks:

- Put the emphasis on elliptic curves, but take a more general view from time to time;
  + the next case; sometimes, hec's yield info on ec's.
- Consider any base field, with some preference for large prime fields, or $\mathbb{F}_2$; few places where it really matters.

II. Point counting algorithms: basic approaches

F. Morain

III. Point counting algorithms: elaborate methods

F. Morain

III. Discrete log algorithms

F. Morain

Bibliography and links

- A course in algorithmic algebraic number theory (Cohen);
- The arithmetic of elliptic curves (Silverman);
- Elliptic curve public key cryptosystems (Menezes);
- Elliptic curves in cryptography (Blake, Seroussi, Smart);
- Lercier's thesis.
- Algebraic aspects of cryptography (Koblitz, appendix on hec by Menezes, Wu, Zuccherato).

www.lix.polytechnique.fr/Labo/
(Francois.Morain, Pierrick.Gaudry, Mireille.Fouquet)
cristal.inria.fr/~harley/
I. Elements of theory.

Let $C$ be a plane smooth projective curve of genus $g$ with equation $F(X,Y) = 0$ with coefficients in $\mathbb{K}$.

Conic: (genus 0) $x^3 + y^2 = 1$.

Elliptic curve: (genus 1) $y^2 = x^3 + ax + b$.

Hyperelliptic curve: (genus $g$) $y^2 = x^g + \cdots$ (or in some cases $y^2 = x^{g+1} + \cdots$).

Rem. To simplify things, we assume that $C$ is “at most” hyperelliptic (no $C_4$ or $X_0(N)$).

Def. $C(\mathbb{K}) = \{ P = (x,y) \in \mathbb{K}^2 : F(x,y) = 0 \}$.

Thm. When $g \leq 1$, there is a group law on $C(\mathbb{K})$. When $g > 1$, there is a group law on the jacobian of the curve.

Elliptic curves

$E : Y^2 + a_1XY + a_3Y = X^3 + a_2X^2 + a_4X + a_6$

$b_2 = a_1^2 + 4a_3$, $b_4 = 2a_4 + a_1a_3$, $b_6 = a_1^3 + 4a_1a_3$,

$c_4 = b_4^2 - 27b_6$, $c_6 = b_6^2 - 4b_4b_6$,

$\Delta = -b_2b_6 - 8b_4^3 - 27b_6^2 + 9b_2b_4b_6 \neq 0$

$j(E) = \frac{c_4}{\Delta}$

When $p = 2$: $Y^2 + XY = X^3 + a_2X^2 + a_6$.

$j = 1/a_6$.

When $p > 3$: $Y^2 = X^3 + AX + B$.

$\Delta = -16(4A^3 + 27B^2)$.

$E(\mathbb{K})$, tangent-and-chord ($\oplus$, $O_E$), multiplication by $n$ noted $[n]P.$
Hyperelliptic curves

\[ y^2 + h(x)y = f(x) = x^{2p+1} + \cdots \]

Representing \( \mathcal{J}_g(C) \): (Mumford)
An element \( \langle u, v \rangle \) of \( \mathcal{J}_g(C) \) is
\[ D \leftarrow \langle u(x), v(x) \rangle, \quad \text{deg}(u) \leq g, \quad \text{deg}(v) < \text{deg}(u), \]
defined by \( (P) = (x_i, y_i) \).
\[ u(x) = \prod_{i=1}^g (x - x_i), \quad \text{and} \quad v(x) = y_i, \quad \forall i. \]

Rem. If \( D \leftarrow \langle u, v \rangle \), then
\[ -D \leftarrow \langle u, v \rangle, \quad \text{then} \]

Group law: Cantor's algorithm (or special formulae for fixed \( g \) à la Spallek, Harley, Nagao).

Cardinality
\[ \mathcal{K} = \mathbb{P}_g = \mathbb{P}_v : N_c = \# \mathcal{J}_g(C) \quad \text{where} \quad |\mathcal{K} : \mathbb{K}| = r: \]
\[ Z(T) = \exp \left( \sum_{n \geq 1} \frac{T^n}{n} \right). \]
Ex. \( \mathbb{P}_1(\mathbb{P}_n) = \{(x, z) \neq (0, 0) \in \mathbb{P}_n^2 \} / \sim. \)
\[ \# \mathbb{P}_1(\mathbb{P}_n) = 1 + q^n \]
\[ Z(T) = \frac{1}{(1 - T)(1 - qT)} \]
Thm. (Weil) \( Z(T) \in \mathbb{Q}[T] \)
\[ Z(T) = \frac{L(C)}{(1 - T)(1 - qT)} \]
(i) \( L(T) = 1 + a_1T + \cdots + q^gT^g \quad a_i \in \mathbb{Z}; \)
(ii) \( a_{g+1} = q^{g+1}a_1 \quad \text{for} \quad 0 \leq i \leq g; \)
(iii) \( L(T) = \prod (1 - a_iT), \quad \text{then} \quad a_0, a_{g+1} = q \quad \text{and} \quad a_1 = \sqrt[q]{a} \)
Thm. \#\mathcal{J}_g(C) = L(1).
Cor. \#\mathcal{J}_g(C) = (g + 1)! \leq 2 \sqrt{g}
\[ (\sqrt[q]{a} - 1)^{2g} \leq \#\mathcal{J}_g(C) \leq (\sqrt[q]{a} + 1)^{2g}. \]

\( \ell \)-torsion
Det. \( \mathcal{J}_g(p) = \{ P \in \mathcal{J}_g(K) | [n] P = \mathcal{O}_g \} \)
Thm. \# \mathcal{J}_g(p) = \# \mathcal{J}_g(K)/\mathcal{O}_g, \quad n = 1, \mathcal{J}_g(p) \sim \mathbb{Z}/n\mathbb{Z}; \)
\( \mathcal{J}_g(p^k) = \mathbb{Z}/p^{k}\mathbb{Z}, \quad 0 \leq k \leq g. \)
Rem. In general \( k = g \) (ordinary curves); when \( g = 1 \), the case \( k = 0 \) corresponds to supersingular curves.
Cor. \#\mathcal{J}_g(C) \mathbb{K} \text{ is at most } C_1 \times C_1 \times \cdots \times C_{2g}.

Division polynomials for elliptic curves:
Take \( E : y^2 = x^3 + Ax + B \):
\[ [n](X, Y) = \left( \frac{\phi_n(X, Y)}{\psi_n(X, Y)} \right)^{\omega_n(X, Y)} \]
\[ \phi_n - X \psi_n^2 - \psi_{n+1}\psi_{n-1} \]
\[ 4Y \omega_n - \psi_{n+2}\psi_{n-2} - \psi_{n}\psi_{n+2} \]
\[ \phi_n \psi_{2n+1} \psi_{2n}/(2Y), \omega_{2n+1}/Y, \omega_{2n} \in \mathbb{Z} \{ A, B, X \} \]
Rem. When \( g > 1 \), one can define analogous division polynomials (cf. Cantor).
**II. Generic methods**

**Input:** a finite abelian group $(G, +)$ with $|G| \leq B$.

**Output:** $|G|$.

**Enumeration:** $O(|G|)$. Use Lagrange’s theorem: for random $z \in G$, find $\omega = \text{order of } z$. Deduce from this the order of $G$ (take care to small orders, group structure with SNI, etc.; see Cohen).

**How to find $\omega$:**
- try increasing value of $\omega: O(\omega) \leq O(B)$ (??).
- Shanks’ baby steps/giant steps method: write $m = m_0 + m_1 b$ for some $b$. $0 \leq m_1 < b$, $0 \leq m_1 \leq B/b$ and write
  \[ [m]_b z = 0 \iff [m_1] \left( [b] z \right) = [m_0] z. \]

1. **baby steps:** precompute $B = \{ [m_1]_b z, 0 \leq m_1 < b \}$;
2. **giant steps:** find all $m_1$ s.t. $[m_1] \left( [b] z \right) = [m_0] z$ for some $m_0$.

---

**Application to elliptic curves**

- **Enumeration:** find all $z \in \mathbb{F}_p$ s.t. $f(z)$ is a square.
- **Lagrange:** $[q + 1] P = [\pm c] P$ for $0 \leq c \leq 2 \sqrt{q}$.
  Rem. If $\text{gcd}(P)$ is large enough, then
  \[ \# \{ c \in [-2 \sqrt{q}, 2 \sqrt{q}], [q + 1 - c] P = Ox \} = 1 \]
  and we can bypass the structure problem.
- **Kangaroos:** idem.
- **Shanks:**
  More precisely: $c = n_0 + n_1 W$, $0 \leq n_0 < W$.
  \[ \text{Cost: } W = \sqrt{\text{gcd}}(N, J), \text{ so } O(\sqrt{\text{gcd}}). \]

---

**Rem.** membership must be tested very fast (hashing).

**Rem.** can be modified when $A \leq |G| \leq B$, yielding a method in $O(\sqrt{B - A})$.

- Using kangaroos (Stein-Teske, Gaudry-Harley): probabilistic method in $O(\sqrt{B - A})$ time and $O(1)$ space (see discrete logs in part III).
Application to hyperelliptic curves

\[ L(\ell) = 1 - r_1 + \cdots + (-1)^{r_d} r_d \]
\[ + (-1)^{d+1} q r_{d+1} - \cdots - q^{r_d} r_d q^d, \]
\[ |s_1| \leq \left( \frac{2g}{g} \right)^{q^{r_d}/2}. \]

A) Enumeration

\( g = 2 \): compute \( N_1(C) \) and \( N_2(C) \) and deduce
\[ s_1 = q + 1 - N_1(C), \]
\[ s_2 = (q^2 + 1)/2. \]

\( g = 3 \): set \( s_3 = (q^3 - 3q^2 + 2)/3. \)

Prop. Method in \( O(q^{(g-1)/2}) \).

B) Lagrange

Hasse-Weil gives
\[ w \sim \left( \frac{q}{q-1} \right)^{2g} (1 - \left( \frac{q}{q-1} \right)^{2g} - 4qg^{1/2} + O(q^{-1/2})) \] (for fixed \( g, q \rightarrow \infty \)).

Prop. Method in \( O(q^{3g/2}) \) (for fixed \( g \)).

III. Particular curves

A) Supersingular curves

Supersingular elliptic curves: \( E \) s.t.
\[ \#E = q + 1 - c \cdot p \] (not every \( c \), all is known).

For instance: when \( n = 2m + 1, q = 2^n \)

<table>
<thead>
<tr>
<th>( E )</th>
<th>( c )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y^2 + Y = X^3 + X )</td>
<td>0</td>
</tr>
<tr>
<td>( Y^2 + Y = X^3 + X )</td>
<td>(-2/n) ( \sqrt{q} )</td>
</tr>
<tr>
<td>( Y^2 + Y = X^3 + 1 )</td>
<td>( 2/n) ( \sqrt{q} )</td>
</tr>
</tbody>
</table>


\( \text{Pb:} \) subject to the MOV reduction (see also Frey, Rück).

\( g > 1 \): can be generalized, but reductions still apply (see also Galbraith for security evaluation).

B) Weil-Koblitz

Build curves over \( \mathbb{F}_q \) for \( q \) small and use \( \text{Jac}(C)/\mathbb{F}_q \).

\( \text{Pb:} \) ECDL might be a little easier.

C) Shanks/Kangaroos

Prop. Method in \( O(q^{3g/2}) \) (for fixed \( g \)).

Using partial information

Prop. (Elkies) If the \( k \) first \( s_j \)’s are known, then
\[ w' = 2 \left( \frac{2g}{k} \right)^{q^{(k-1)/2}} + O(q^{-(1/2)}) \]
and with more work
\[ w'' = \frac{4g}{k} q^{(k-1)/2} + O(q^{-(k-1)/2}). \]

Cost: \( O(q^{(k-1)}) + O(q^{(3g-1)/k}) \) leading to \( k = (2g + 4)/5 \) and \( O(q^{(3g-1)/2}) \).

Ex. \( g = 3 \) leads to \( O(q) \) method by computing \( s_1 \)

Stein-Teske-Williams: use truncated Euler product of the \( L \)-function to approximate \( w \).

Ex. \( g = q = 3 \), \( \text{Jac} \equiv 10^{20} \) in 10 hours on a Pentium Pro/200 with SIMATH.

C) CM curves

\( \text{complex multiplication} \)

\( g = 1 \):

Thm. (Katz) if \( p = z^2 + 4g^2 \) with \( z \equiv 1 \mod 4 \) and \( a \equiv 0 \mod p \), then \( E : Y^2 = X^3 + aX \) has cardinality
\[ p + 1 = \begin{cases} 2z & \text{if } (a/p) = 1, \\ -2z & \text{if } (a/p) = -1, \\ -4g & \text{otherwise with } y \text{ s.t. } 2y(a/p) = z. \end{cases} \]

There are 13 cases of curves defined over \( \mathbb{Q} \) having such properties; in general, \( 4p = A^2 + DB^2 \), \( \#E = p + 1 - A \) basis for primality proving with elliptic curves (ECPP, Atkin, M.).

\( g > 1 \):

Spallek, Weng (\( g > 2 \); Buhler-Koblitz; Duursma-Sakurai; Chao, Matsuda, Nakamura, Tsujii).

\( \text{Pb:} \ too \ too \ structure? \)

D) Weil descent

Start from ec’s to build hec’s (Smart et al.).
IV. Schoof’s algorithm

The Frobenius endomorphism

Ordinary:

\[ \phi : \mathbb{F}_p \to \mathbb{F}_p \]

\[ z \mapsto z^q \]

Extension to \( C \) and \( \text{Jac}(C) \):

\[ \psi : \mathbb{C}(\mathbb{F}_p) \to \mathbb{C}(\mathbb{F}_p) \]

\[ (X, Y) \mapsto (X^q, Y^q) \]

Fundamental thm. The minimal polynomial \( \chi(T) \) of \( \psi \) is the reciprocal of \( E(T) \). Moreover \( \#\text{Jac}(C)/\mathbb{F}_q = \chi(1) \).

Consequence: computing \( \#\text{Jac}(C)/\mathbb{F}_q \) boils down to computing \( \chi(T) \).

\[ g = 1: \text{for } E \text{ with } \chi(T) = T^2 - cT + q, |c| \leq 2 \sqrt{q}. \]

\[ \psi \text{ restricted to } \mathbb{F}_q \text{ satisfies:} \]

\[ \psi^2 - c\psi + q \equiv 0 \mod \ell \]

so we can find \( c \equiv r \mod \ell \) such that

\[ (X^q, Y^q) \oplus [c] (X^q, Y^q) \]

in \( \mathbb{K}[X, Y]/(E, f_E(X)) \) and use CRT once

\[ \ell > 4 \sqrt{q} \]

Yields a \( O(\log^2 q) \) deterministic algorithm.

Pb. \( \deg(f_E) = O(\ell^2) \).

\[ g > 1: \text{general algorithm by Pila (1990), but impossible} \]

\[ \text{to implement; Kampk"{o}tter (1991) for any hyperelliptic,} \]

\[ \text{with precise equations for } g = 2 \text{ (uses Gr"{o}bner bases).} \]

---

II. Point counting algorithms:

elaborate methods

F. Morain

http://www.lix.polytechnique.fr/Labo/
Francois.Morain/
I. What we saw yesterday

\[ \varphi : C(F) \to C(F) \]

\( (X, Y) \mapsto (X^q, Y^q) \)

Fundamental thm. The minimal polynomial \( \chi(T) \) of \( \varphi \) is the reciprocal of \( E(T) \). Moreover

\[ \#\text{Jac}(C)/\mathbb{F}_q = \chi(1). \]

Consequence: computing \( \#\text{Jac}(C)/\mathbb{F}_q \) boils down to computing \( \chi(T) \).

\[ g = 1 : \text{for } E \text{ with } \chi(T) = T - cT + q, |c| \leq 2 \sqrt{q}, \]

\( \varphi \) restricted to \( E[F] \) satisfies:

\[ \varphi^1 \circ \varphi + q \equiv 0 \pmod{\ell} \]

so we can find \( c \in \mathbb{F}_q \) such that

\[ (X^q, Y^q) \circ \varphi (X, Y) = [q](X^q, Y^q) \]

in \( \mathbb{F}[X, Y]/(E, f_0(X)) \) and use CRT once

\[ \prod \ell > 4 \mathbb{F} \text{ yields } O(\log^3 q) \text{ deterministic algorithm.} \]

Pb. \( \deg(f_0) = O(\ell^2) \).

II. Isogenies and point counting

A) Elements of theory

Def. \( O : \mathbb{F} \to \mathbb{F} \)

\( O(X, Y) = \varphi(X, Y) \)

or \( \varphi \) induces a morphism of groups.

First examples

1. \( \{E(X, Y) = (X, Y) \} \) on \( E : Y^2 = X^3 - X \).
2. \( \{E(X, Y) = (-X, Y) \} \) on \( E : Y^2 = X^3 - X \).
3. \( \varphi(X, Y) = (X^q, Y^q) \), \( \mathbb{F} = \mathbb{F}_q \).

Thm. (dual isogeny) There is a unique \( \tilde{\varphi} : E^* \to E \), \( \tilde{\varphi} \circ \varphi = [m] \), \( m = \deg(\varphi) \).

\[ E \]

\( \tilde{\varphi} \)

\[ E^* \]

\[ [m] \]

\[ \varphi \]

\[ \tilde{\varphi} \]

Application to point counting

Suppose \( F \) is a subgroup of order \( \ell \) of \( E \):

\[ E \]

\( I \)

\( E^* \)

\[ [\ell] \]

\[ I \]

\[ \ell \]

\[ E \]

\( I(X, Y) = \left( \frac{G(X)}{H(X)} \right), \deg(H) = (\ell - 1)/2 \)

\( \text{let } [\ell] \subseteq E[\ell] \Rightarrow H(X) \mid f_0(X) \in \mathbb{F}[X] \).

Schoof’s algorithm on a degree \( O(\ell) \) polynomial.

Pb. When does such an \( F \) exist over \( \mathbb{F} \)?

Slide 25

Isogenies and subgroups

Thm. If \( F \) is a finite subgroup of \( E \), then there exists \( \varphi \) and \( E^* \) s.t.

\[ \varphi : E \to E^* = E/F, \quad \ker(\varphi) = F. \]

Ex. \( E : y^2 = x^3 + ax^2 + bx, F = \langle (0, 0) \rangle \)

\( E^* : Y^2 = X^3 - 2aX^2 + (a^2 - 4b)X, \)

\( \varphi : (x, y) \mapsto \left( \frac{y^3}{x^2}, \frac{y(b - x)}{x^2} \right) \).

More generally: Vélu’s formulas give

\[ \varphi(X, Y) = \left( \frac{G(X)}{H(X)} \right), \quad J(X, Y) \]

(case \( \deg(\varphi) \) odd.)

Slide 26

Application to point counting

Suppose \( F \) is a subgroup of order \( \ell \) of \( E \):

\[ E \]

\( I \)

\( E^* \)

\[ [\ell] \]

\[ I \]

\[ \ell \]

\[ E \]

\( I(X, Y) = \left( \frac{G(X)}{H(X)} \right), \deg(H) = (\ell - 1)/2 \)

\( \text{let } [\ell] \subseteq E[\ell] \Rightarrow H(X) \mid f_0(X) \in \mathbb{F}[X] \).

Schoof’s algorithm on a degree \( O(\ell) \) polynomial.

Pb. When does such an \( F \) exist over \( \mathbb{F} \)?

Slide 27
B) Atkin and Elkies

Consider \( \psi : (X, Y) \mapsto (X, Y^n) \) and its restriction \( \psi_f \) to \( E[F] \):

\[
\psi_f^2 - c\psi_f + q = 0,
\]

\( \Delta = c^2 - 4q \).

If \( (\Delta/\ell) = -1 \) then over \( \mathbb{Z}_\ell \),

\[
\text{Mat}(\psi_f) \cong \left( \begin{array}{cc} \lambda_1 & 0 \\ 0 & \lambda_2 \end{array} \right) \Rightarrow \exists F, \psi(F) = F \Rightarrow F
\]

is a cyclic subgroup of order \( \ell \), defined over \( \mathbb{K} \).

Con. If \( (\Delta/\ell) = 1 \), \( j_F \) has a factor of degree \( (\ell - 1)/2 \).

Pb. How do we know that \( (\Delta/\ell) = 1 \)?

---

Modular polynomials

Thm. \( \exists \Phi_f(X, Y) \in \mathbb{Z}[X, Y] \) s.t. \( E \) and \( E' \) are isogenous over \( \mathbb{K} \) only if \( \Phi_f(j(E)), j(E') = 0 \).

Rem. This polynomial comes from the theory of elliptic curves over \( \mathbb{C} \).

Ex.

\[
\Phi_2(X, Y) = X^3 + X^2(-Y^2 + 1488Y - 162000)
\]

\[
+X(1488Y^2 + 4073375Y + 874900000)
\]

\[
+Y^3 - 162000Y^2 + 874900000Y - 15746400000000000.
\]

Thm. \( E/\mathbb{F}_\ell \):

\[
\Phi_f(X, j(E)) = \begin{cases} \frac{1}{(\ell-1)} \{1\}^{(\ell-1)} & \text{if } (\Delta/\ell) = 1, \\
\{e\}^{(\ell-1)} & \text{if } (\Delta/\ell) = -1 
\end{cases}
\]

and \( e \) is the order of \( \lambda_1/\lambda_2 \).

Atkin's idea: use the splitting of \( \Phi_2 \) to deduce information on \( \ell \) and combine it via a clever match and sort algorithm.

---

Elkies's algorithm

Repeat

1. factor \( \Phi_f(X, j(E)) \) over \( \mathbb{K} \).
2. if type = \( (1)(1)[\ell] \ldots [\ell] \):
   2.1 build \( E' \);
   2.2 build \( I_1 \);
   2.3 find \( c \mod \ell \);
until \( \prod_{\ell \neq \text{good}} \ell > 4\sqrt{\ell} \).

Thm. \( O(\log^3 q) \) probabilistic.

Rem. One can build \( \Phi_f \) using series operations: \( \Phi^c, \Phi^p \).

\( \Phi^c \) (Atkin), \( \Phi^p \) (Müller), for \( \ell \leq 1000 \).
More details

- \( p \gg \varepsilon \) (Elkies, Atkin, 1988–1992)
  - use the theory of elliptic curves and lattices over \( \mathbb{C} \) (Weierstrass \( \wp \) function); rational formulas for \( E^* \);
  - computing \( I \) takes \( O(\ell) \) steps and uses \( O(\ell) \) memory;
  - very efficient in practice: \( p \) with 500 decimal digits (M. 1995).
- \( p \) small: Couveignes I (first efficient method, 1994),
  - Crit.

### C) Using formal groups

\[ t = -X/Y, s = -1/Y \]

\[ F(t, s) := t^3 + a_1 t s + a_2 t^2 s + a_3 s^2 + a_4 t s + a_6 s^3 - s = 0, \]

\[ s = S(t) = t^3 + a_1 t + (a_1^2 + a_2) t^2 + O(t^3). \]

\[ Y = \frac{1}{S(t)} - t^{-3} a_1 t^{-2} + a_2 t^{-1} + a_3 + (a_1 a_2 + a_4) t + O(t^2) \]

\[ X = \frac{1}{S(t)} - t^{-2} - a_2 t^{-1} + (a_1 a_2 + a_4) t^2 + O(t^2) \]

\( \ell_1 = \ell_1(t_1, t_2) = t_1 + t_2 - a_2 t_1 - a_3 t_2 + a_1 t_1 t_2 + t_1 t_2 \)

\( \phi = \psi \ell_1 t_1 t_2 - 3a_3 \ell_1^2 t_1^2 + 2a_2 t_1 \ell_1 + \ldots \)

\[ [\ell_1] t_2 = 2 t_1 - a_3 t_1^2 + a_2 t_1^3 + (a_1 a_2 - 7 a_3) t_1^4 + \ldots \]

\[ r_n(t_1) = n t_1 + \ldots \]

**Thm.** \( \phi(t) = \Psi_2(e(t))^2 = c_\ell(E) \ell^2 + O(\ell^3) \).

**Thm.** Norm\( E_{/\mathbb{Q}}(\ell) \approx \frac{1}{\tau_\ell^2} (\ell^2 - 1) \pmod{\ell} \).

**Ex.** \( c_\ell([a_1, a_2, a_3, a_4, a_6]) = a_1, c_\ell([a_1, a_2]) = a_2. \)

### Couveignes

Looking for \( I \) is equivalent to searching for a morphism of formal groups \( \mathcal{M} : E \to E^* \)

\[ \mathcal{M}(\ell(t)), \phi(t)] \to [\mathcal{U}(\ell(t)), \phi^*(\mathcal{U}(\ell(t)))] \]

s.t.

\[ \mathcal{M}(P_1 \oplus P_2) = \mathcal{M}(P_1) \oplus \mathcal{M}(P_2) \]

or

\[ \mathcal{U}(\ell(t_1, t_2)) = \mathcal{U}(\ell(t_1)) \oplus \mathcal{U}(\ell(t_2)) \]

where \( \mathcal{U}(\ell(t)) = t + \sum_{i \geq 1} u_i t^i. \) \( \mathcal{U} \) is found incrementally using indeterminate coefficients.

**Thm.** time \( O(\ell^3) \), space \( O(\ell^2) \).

**Rem.** In real life, difficult to implement (huge series computations).

For more details, see Lercier & Morain, Math. Comp. 69, 2000.

### D) Lercier’s algorithm

**Motivation:** Couveignes I too slow, too much memory involved (huge series computations).

\( p = 2, E : Y^2 + X Y = X^3 + a_1 E^* : Y^2 + X Y = X^3 + b \)

**Prop.** \( I(X) = X L(X)^2 / H(X)^2 \).

**Ideas:**

- send \( P = (0, \sqrt{b}) \) to \( P^* = (0, \sqrt{b}) \) and use:
  \[ P \oplus M = (0, \sqrt{b}) \oplus (X, Y) = (\sqrt{b} X, \ldots) \]

- use
  \[ I(P \oplus M) = P^* \oplus I(M) \]

- use
  \[ I = [2] = [2] + I \]

- translate all these equations in a (non linear) system of equations in coefficients of \( L_1(X) \) and \( H(X) \) and solve it.

---

**Slide 35**

- **Prop.** More details

- **Prop.** C) Using formal groups

- **Prop.** Couveignes

- **Prop.** D) Lercier’s algorithm
Analysis: heuristically, time is $O(t^2)$ with a much better constant; space is $O(t)$.

In practice: considerably faster than Couveignes’s first algorithm: 400 times faster for computing isogenies for \( \mathbb{F}_{q^{215}} \), resulting in a speed-up of 5 on the total running time for the complete SEA algorithm.

<table>
<thead>
<tr>
<th>Field</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mathbb{F}_{q^{215}} )</td>
<td>80 sec</td>
</tr>
<tr>
<td>( \mathbb{F}_{q^{232}} )</td>
<td>308 sec</td>
</tr>
<tr>
<td>( \mathbb{F}_{q^{246}} )</td>
<td>2.861 sec</td>
</tr>
<tr>
<td>( \mathbb{F}_{q^{251}} )</td>
<td>3.300 sec</td>
</tr>
</tbody>
</table>

(Alpha 266 MHz)

Rem. Vercauteren (October 1999) \( \mathbb{F}_{q^{246}} \) after 65 days of CPU time on 400 MHz PCs.

Use interpolation: let \( A(X) \ s.t. \ A(x_m) = x_m^* \) for all \( m, 1 \leq m \leq d_k \). Then

\[
I(X) = A(X) \mod f_p(X)
\]

Prop. If \( d_k > 2t \), recover \( H \) via the euclidean algorithm.

Rem. In general, the \( p^k \) division is not rational and the interpolation part must be done with some care.

Prop. Complexity should be time \( O(t^{3t}) \), space \( O(t^2) \).

Prop. Artin-Schreier approach of Couveignes: \( O(n, M(t^2)), \) see Math. Comp. 69, 2000.

### E) CouveignesII

Idea: \( E[p] \) seems to be of first importance.

Prop.

\[
\begin{array}{ccc}
\hat{P} & \psi & P \\
\uparrow & & \downarrow \\
\bar{P} & & \bar{P}
\end{array}
\]

\[
[p](X, Y) = (f_p(X), G_p(X, Y))
\]

Coro.

\[
f_p(X) = \bar{f}_p(X), \ \deg \bar{f}_p = d_p = p^e - 1 \pmod{2} \Rightarrow 2^{e-1}
\]

Idea: \( I \) maps \( E[p^e] \) to \( E^{[p^e]} \);

\[
I([m]P) = [m]P^e, \ I(x_m) = x_m^e
\]

### III. Satoh’s algorithm

**Det.** \( \mathbb{Z}_p \) ring of \( p \)-adic integers \( (x_0, x_1, \ldots, x_n, \ldots) \) s.t. \( x_n \in \mathbb{Z}/p^n \mathbb{Z} \) and \( x_{n+1} = x_n \mod p^n \). Denote by \( \pi : \mathbb{Z}_p \to \mathbb{Z}_q \) sending \( x \) to \( x \).

**Det.** Let \( q = p^e \) and \( f(t) \in \mathbb{Z}[t] \) s.t. \( \pi(f) \) is irreducible in \( \mathbb{F}_q[t] \). Then \( \mathbb{Z}_q \to \mathbb{Z}_q/f(t) \).

An element of \( \mathbb{Z}_q \) is \( A = a_0 \cdot t^{e-1} + \cdots + a_n \) with \( a_i \in \mathbb{Z}_q \); \( \mathbb{Z}_q \) contains \( \mathbb{Z}_p \) as a subring.

\[
\pi(A) = \sum \pi(a_i) t^i
\]

Prop. Let \( \sigma \) be the little Frobenius sending \( x \) in \( \mathbb{Z}_q \) to \( x^p \). There is a canonical way to lift \( \sigma \) to \( \Sigma : \mathbb{Z}_q \to \mathbb{Z}_q \).

Extend \( \sigma \) to points \( \sigma(x, y) = (\sigma(x), \sigma(y)) \) and to curves: \( \sigma(E) = \{ \sigma(x), \sigma(y) \} \), so that if \( P \in E(\mathbb{Z}) \), then \( \sigma(P) \in \sigma(E)(\mathbb{Z}) \).
**Thm.** (Lubin-Serre-Tate) Let $E / \mathbb{Q}$ with \( j = j(E) \in \mathbb{Q} - \mathbb{Q}^* \). There is a unique $J$ in $Z_4$ s.t.
\[
\Phi_2(J, \Lambda(J)) = 0,
\]
\( \pi(J) = j \), $J$ is the invariant of the canonical lift $E$ of $E$ and $End(E) = End(E)$.

**Isogeny cycles:**
\[
\begin{align*}
\varepsilon_0 & \xrightarrow{\pi} \varepsilon_1 \xrightarrow{\pi} \cdots \xrightarrow{\pi} \varepsilon_{2g-2} \xrightarrow{\pi} \varepsilon_0 \\
\downarrow & \downarrow \quad \downarrow \quad \downarrow \\
E_0 & \xrightarrow{\sigma_{2-1}} E_{2g-1} \xrightarrow{\sigma_{2-1}} \cdots \xrightarrow{\sigma_{2-1}} E_1 \xrightarrow{\sigma_{2-1}} E_0
\end{align*}
\]

**Prop.** $\varphi = \sigma_0 \circ \sigma_1 \circ \cdots \circ \sigma_{2g-1}$

**Thm.** $\text{Tr}(\varphi) = \text{Tr}(\varphi)$. Slide 41

---

**Computing $\text{Tr}(\mathcal{F})$**

Use the dual of Frobenius to get another isogeny cycle amenable to computations:
\[
\begin{align*}
\varepsilon'_0 & \xrightarrow{\downarrow \pi} \varepsilon'_1 \xrightarrow{\downarrow \pi} \cdots \xrightarrow{\downarrow \pi} \varepsilon'_{2g-2} \xrightarrow{\downarrow \pi} \varepsilon'_1 \\
\downarrow & \downarrow \quad \downarrow \quad \downarrow \\
E'_0 & \xrightarrow{\sigma'_{2-1}} E'_{2g-1} \xrightarrow{\sigma'_{2-1}} \cdots \xrightarrow{\sigma'_{2-1}} E'_1 \xrightarrow{\sigma'_{2-1}} E'_0
\end{align*}
\]

**Prop.** $\varphi' = \sigma'_{2g-1} \circ \sigma'_{2g-2} \circ \cdots \circ \sigma'_{1}$ (ident for $\mathcal{F}'$) and also $\text{Tr}(\mathcal{F}) = \text{Tr}(\mathcal{F}) = \text{Tr}(\varphi')$.

Let $\tau$ (resp. $\tau_1$) denote the local parameter of $E$ (resp. $E_1$).
\[
\mathcal{F}(\tau) = \sum_{k \geq 1} \tau^k,
\]

**Prop.** (Satoh) $\text{Tr}(\mathcal{F}) = c_1 + q/c_1$.
\[
c_1 = \prod_{i=0}^{2g-1} d_i
\]

where (Vélu’s formulas again)
\[
\Sigma_i (c_i) = g_i \tau_1 + O(\tau_1^2)
\]

---

**Satoh’s algorithm in brief**

1. Compute the curves $E_0, E_1, E_{2g-1}$ and their invariants $j_i$.
2. Lift all the $j_i$’s simultaneously by a Newton iteration to get $J_i$:
\[
\Theta(x_i) - (\Phi_i(x_i, \varepsilon_i), \Phi_i(x_i, \varepsilon_i), \ldots, \Phi_i(x_i, \varepsilon_i))
\]
as
\[
(x_i) = (x_i) - ((D\Theta)^{-1}(\Theta)(x_i))
\]
3. Lift each $E_i$ coefficient by coefficient.
4. Lift the $p$-torsion subgroup of $E_i$.
5. Compute the $\varepsilon_i$’s.
6. Compute the trace.

**Thm.** (Satoh-FGH) For fixed $p$, Satoh-FGH requires $O(e^3)$ memory and $O(e^{3+})$ bit-operations.

---

**Results**

**Implementation details:** too many of them, see [FGH] and the URL
http://www.xent.com/~harley
for a demo version of the ECPC program.

**Record:** $\varepsilon^{22}$ in 313 hours on an Alpha EV6, 750 MHz, 16.9 Gb in Cornell.

**More timings:** on Alpha 750 MHz

<table>
<thead>
<tr>
<th>Degree</th>
<th>Time</th>
<th>Degree</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>160</td>
<td>2.5 s</td>
<td>1000</td>
<td>22m</td>
</tr>
<tr>
<td>200</td>
<td>4.6 s</td>
<td>2000</td>
<td>3A</td>
</tr>
<tr>
<td>240</td>
<td>7.1 s</td>
<td>4000</td>
<td>29A</td>
</tr>
</tbody>
</table>
IV. Generalization to genus 2

- Division polynomials: Cantor.
- Schoof: Pila; Kampschiter (Grobner bases).

Modular equations:
- Siegel’s modular forms; large output. On-going work by Gaudry and Harley.
- Alternative equations by Gaudry and Schost (similar to Charlap, Coley and Robbins for elliptic curves): computed for $g = 2, \ell = 3$ (see PG’s web page for the equation).
- Factorization patterns: exist; cf. PG for $\ell$.

Isogenies: Velu’s formulas?

Satoh’s algorithm: LST valid. Need modular equation.

Current status

- **Genus 2**: Stein-Teske (1999): $g = 10^8 + 7$, $g = 3$, $\#J_\mathbb{C} = 10^{26}$ in 10 hours on a Pentium Pro200 with SIMATH.
- **Genus 2**: Gaudry & Harley (ANTS IV)
  - find $\#J_\mathbb{C} \mod \ell$ using Cartier-Manin ($\ell = p$) or Cantor’s division polynomials;
  - find $\#J_\mathbb{C} \mod 2^\ell$ by looking for solutions of $[2]D = D_2$;
  - combine all this to reduce the birthday paradox.
  - record: $p = 10^{18} + 51$:
    $$y^2 = x^5 + 31415926535 \cdot 9970333 \cdot z^4 + \cdots$$
    $$\#J_\mathbb{C} = 9999999999992, a = 712006714522770002, g = 100000000,$$

<table>
<thead>
<tr>
<th>$\ell^2$</th>
<th>2^16</th>
<th>3</th>
<th>5</th>
<th>7</th>
<th>11</th>
<th>13</th>
</tr>
</thead>
<tbody>
<tr>
<td>time</td>
<td>10A</td>
<td>1200 A</td>
<td>300 A</td>
<td>12 A</td>
<td>10 A</td>
<td>26 A</td>
</tr>
</tbody>
</table>

Width 2.5 $\times 10^9$, 50 days on a 500 MHz Alpha.

V. Generating cryptographically strong curves

$\mathbb{F}_p$ with large $p$ or $\mathbb{F}_q$, with $n$ prime (Weil descent, see Menezes & Qu); subgroups of large prime order.

- **Supersingular curves**: too much structure (?).
- **CM curves**: quite efficient for $g = 1$ or $g = 2$, but who knows?
- **Fixed curves**: The NIST curves (?).
- **Random curves**:
  - $g > 1$: not efficient yet.
  - $g = 1$: use SEA for large $p$; Satoh for $p = 2$.

Very efficient when combined to the early-abort approach in Lercier’s EUROCRIPT’97 article. Experiments conducted by FGH combining SEA and Satoh show that it takes 5 min on Alpha 750 MHz to build a good curve over $\mathbb{F}_{2^{151}}$.

Slide 46

II. Discrete log algorithms

2nd MAGC Workshop

November 19, 2000

F. Morain

Slide 47

Slide 48
Plan

I. The problem.

II. Generic methods.

III. Index calculus.

IV. Cryptographic implications.

I. The problem

$K = \mathbb{F}_q = \mathbb{F}_r$.

Input: $P \in \text{Jac}(C)$ of order $n$, $Q \in <P>$.

Output: $k$ s.t. $Q = [k]P$.

Ex. elliptic case

$C/K : Y^2 + a_1 XY + a_3 Y = X^3 + a_2 X^2 + a_4 X + a_6$

or hyperelliptic case:

$C/K : Y^2 + H(X)Y = X^{2r+1} + \ldots$


II. Generic methods

A) Shanks

$k \in [-(n - 1)/2, (n - 1)/2], \quad w = \sqrt{(n - 1)/2}$

$k = uv + v, \quad 0 \leq v < w, \quad |k| \leq w.

\[ [k]P = Q \iff [u]([v]P) = Q \oplus [v]P. \]

baby steps: compute the points $Q \oplus [v]P$ for all $v$'s in $O(w)$ steps.

\begin{itemize}
  \item giant steps: find $w$ in $O(w)$ steps.
  \item Cost: $O(w) = O(\sqrt{n})$ in time and space, deterministic.
\end{itemize}

B) Pohlig-Hellman

\[ n = \prod_p p_i^{e_i} \]

\begin{itemize}
  \item Idea: finding $[x]^P = Q$ is equivalent to finding $x \text{ mod } n$, i.e., $x \text{ mod } p_i^{e_i}$ for all $i$ (CRT). It reduces the discrete log mod $n$ to that mod $p_i^{e_i}$ for all $i$.
  \item Cost: $O(\log n \cdot DL(p_i))$.
\end{itemize}

III. Index calculus

General results

\begin{itemize}
  \item No (known) subexponential method for small $g$ (including $g = 1$); recover a subexp method when $g$ increases.
  \item Reduction $\text{Jac}(C)/\mathbb{F}_q \to \mathbb{F}_q$ with $k$ small:
    \begin{itemize}
      \item Supersingular curves: MOV (Menezes, Okamoto, Vanstone using the Weil pairing); Frey & Rück (using the Tate pairing); Galbraith. See G. Frey's talks.
      \item other cases: elliptic curves with $l = 2$ with the Tate pairing.
    \end{itemize}
  \item Discrete logs in subgroups of order $p^f$ of $\text{Jac}(C)/\mathbb{F}_q$, can be found in polynomial time:
    \begin{itemize}
      \item $g = 1$ (anomalous curves) done by Satoh-Araki, Semaev, Smart; $g > 1$ by Rück.
    \end{itemize}
  \item Elliptic curves: lifting points or XEDNI: Silverman et al., Cheon et al. Do not work in the present form.
\end{itemize}
C) Pollard’s $\rho$

Idea: use the properties of functional graphs.

Let $f$ be a random function: $\left< P \rightarrow \L P \right>$.

One starts from $R_0 \in \L P$ and builds $R_{i+1} = f(R_i)$:

\[ R_0 \rightarrow R_1 \rightarrow R_2 \rightarrow \cdots \rightarrow R_{i+1} \rightarrow R_i \rightarrow \cdots \]

More precisely, one starts from $R_0 = [u_0]P \equiv [v_0]Q$ with $u_0$ and $v_0$ in $\mathbb{Z}/n\mathbb{Z}$, and one builds the sequence $R_{i+1} = f(R_i)$, which also builds $[u_i]$ and $[v_i]$ modulo $n$ s.t. $R_i = [u_i]P + [v_i]Q$.

Random mapping statistics: Flajolet & Odlyzko, EUROCRYPT’89: a functional graph for a random $f$ with $n$ vertices should have $0.5 \log n$ components, $\Pi_n = \Pi_0 = \sqrt{\pi n/2}$. 

D) Parallel collision search

1) Distinguished points

Quisquater & Delescaille, EUROCRYPT’89

Idea: store points with a distinguishing property (e.g., $w$)

$\Pi_n = \Pi_0 = \sqrt{\pi n/2}$

Rem. All solved exercises of Certicom done this way.

Choosing $f$

- additive random walk:
  - $\rightarrow$ precompute $r$ points $\{T^{(j)}\}_{0 \leq j < r}$ in $\L P$.
  - $\rightarrow$ $T^{(j)} = [u^{(j)}]P + [v^{(j)}]Q$.
  - $\rightarrow f(R) = R + T^{(j)}$, with $j \equiv H(R)$ where $H$ sends $P$ to $\{0, 1, \ldots , r - 1\}$. In other words: $R_i = [u_i]P + [v_i]Q$, with $u_{i+1} \equiv u_i + u^{(j)}$ mod $n$; $v_{i+1} \equiv v_i + v^{(j)}$ mod $n$.
  - Salter & Schonh: $r \geq 8$; Teske: $r \geq 20$ (Hildebrand’s result on random walks mod $n$).

- multiplicative random walk: (Gallant et al)
  - choose multipliers $\{\mu_j\}_{0 \leq j < r}$.
  - $f(R) = \left[ \prod_{j=0}^{r-1} \mu_j^j \right] R_0$.

\[ \rightarrow \text{requires at least two processors; } \]
\[ \rightarrow \text{computing } \left[ \mu_j \right] R \text{ must be done quickly. } \]

Rem. All solved exercises of Certicom done this way.
E) Adaptation to algebraic curves

Using equivalence classes

Idea: ~ equivalence relation s.t. 
\( \#(\langle P \rangle / \sim) = n/m \). A random walk should find a collision after \( \sqrt{\frac{2}{m}} \) iterations.

Finding ~:

- Wiener, Zuccherato: \( (x, y) \sim (x, -y) \).
- Gallant, Lambert et Vanstone; Harley: use little Frobenius on ABC curves.
- Duursma, Gaudry, M.: automorphism \( g \) of order \( m \) of \( C: S = T \iff y = gT \).

\[ \begin{align*}
\text{Ext.} & & Y^2 = X^3 + aX \text{ with } \mu \{ (x, y) = (-x, y) \} \\
\text{where } i^2 & & = -1 \text{ (a Frobenius).}
\end{align*} \]

Ex2: On \( Y^2 = X^5 - 1 \), \( a(x, y) = \left( \frac{q}{g}x, y \right) \), \( q \zeta_5 = 1 \), \( -\alpha \) has order 10; action on \( J_{BC} \) as:

\[ (z^2 + w_z + w_0, v_z + w_v) \mapsto \left( z^2 + \zeta_5 w_z + \zeta_5^2 w_0, \zeta_5^4 w_z - v_z - w_v \right). \]

Technical details: well defined \( f^2 \); short cycles.

III. Index calculus methods

A) The basic scheme

Western and Miller; Pollard, Adleman, Merkle, etc.

Methods in

\[ L_N[2, \zeta] = \exp \left( \left( \log N \right)^3 \log \log N \zeta \right). \]

Input: \( G = \mathbb{P}_q[1, \log(C)] \) of large genus; \( G < \langle \sigma \rangle \), \( \alpha = g^s \).

Step 1: find the logarithms of the primes (resp. irreducible polynomials, prime divisors) of a factor basis \( J = \{ p_1, p_2, \ldots, p_k \} \).

Step 2: look for \( b \) s.t. \( \alpha^b \) factors over \( J \):

\[ \alpha^b = \prod_{j=1}^k \alpha_j^{a_j} \]
yields

\[ z + b = \sum_{j=1}^k a_j \log g p_j \mod \#G. \]

How do we find the \( \log_d p_i \)?

Choose random integers \( b_i \) for which

\[ g^{b_i} = \prod_{j=1}^k \alpha_j^{a_j b_i} \]
or:

\[ b_i = \sum_{j=1}^k a_j \log g p_j \mod \#G. \]

When enough relations have been gathered, solve the system.

Cost: \( O(L_{G, 1}[1/2, \epsilon]) \) where \( \epsilon \) depends on \( G \) and/or \( \#G \).

Best algorithm so far: \( \alpha \) is NFS \( O(L_{G, 1/3, \epsilon}) \)

(Gordon, Schirokauer).

Record: Joux & Lercier (100 digits, 1999).
B) Discrete log on hyperelliptic curves

- Algorithm ADH from Adleman, DeMarrais, Huang (ANTS I):
  \[ L_{\log p}(1/2, c) \]
  with \( c \leq 2.14 \) and \( \log p \leq (3g + 1)^{0.98} \) (heuristic using Lovorn’s theorem on smooth polynomials); SNF.
- Flassenberg & Paulus: using sieving techniques; experiments with \( e^2 = e^{3g+2} = \cdots \) (Mueller-Stirn-Thiel): proved \( L_{\log p}(1/2, 1.44) \).
- Extensions, proved analysis and optimizations by Enge:
  \[ L_{\log q}(1/2, c)(0) \]
  if \( \theta \log q \leq g \), with \( \max_{c<0} c(\theta) = +\infty \): easier SNF. Smaller \( c \) by Enge and Gaudry.
- Gaudry’s variant: fast, practical, implemented, breaks a lot of proposed systems.

C) Gaudry’s variant

Main features:
- easy to describe and to analyze.
- index calculus using a basis of \( S \)-smooth divisors, i.e. \( J = \{ \langle u(z), v(z) \rangle \in J_{\mathcal{A}}(C) \mid u(z) \text{ is irreducible of degree } \leq S \} \), with \( S = 1 \) for \( g \leq 8 \).
- Finding relations using a random walk.
- Easier linear algebra.
- exponential asymptotic complexity but very practical and breaks a lot of systems proposed by various authors.

The algorithm

**Input:** \( D_1, D_2 \in J_{\mathcal{A}}(C), n = \sigma(D) \) known.

**Output:** A s.t. \( D_2 = [\lambda] \cdot D_1 \).

1. **[Initialisations]** Fix a smoothness bound \( S \) and build the basis \( \mathcal{F} = \{ g_1, g_2, \ldots, g_n \} \). Compute \( T^{(j)} \leftarrow \langle a^{(j)} \rangle D_1 + \langle b^{(j)} \rangle D_2 \) for \( j \in [1..r] \) and set \( R_0 := \langle a_0 \rangle D_1 + \langle b_0 \rangle D_2 \). Put \( k := 1 \).
2. **[Main loop]**
   (a) Look for a smooth divisor:
   \[ j := \mathcal{H}(R_0), R_1 := R_0 + T^{(j)} \]
   \( a_t + b_t \equiv a_t \mod n, b_t \equiv b_t \mod n \).
   until \( R_k := R_1 - \langle a_r \rangle \cdot v_k, v_k(z) \) is \( S \)-smooth.
   (b) Write \( R_1 \) on \( \mathcal{F} \) factor \( a_r(z), v_r(z) \) over \( \mathcal{F} \); store \( R_k = \sum m_k g_i \) and \( a_k = a_1, b_k = b_k \).
   (c) If \( k < \mu F + 1 \), then \( k := k + 1 \), go to 4.
3. **[Linear algebra]** Find \( (Y_k) \) in \( \text{ker}(M) \) over \( Z/nZ \).
4. Return \( \lambda = -\sum a_k y_k / \sum b_k y_k \) mod \( n \).
More properties

Rem. The algorithm has asymptotic exponential complexity, but works very well in practice (see below), since it must factor polynomials of smaller degree than ADH. Provably (and practically) better than Pollard’s ρ for g > 4.

Size of $J_{\text{jac}}$ breakable: $q | J_{\text{jac}} - q^2 = (2g J_{\text{jac}})^2$:

| $|J_{\text{jac}}|/g$ | 4 | 5 | 6 | 7 | 8 |
|-------------------|---|---|---|---|---|
| 10g               | 3.70 | 3.98 | 4.05 | 4.13 | 4.16 |
| 10f               | 3.83 | 3.94 | 4.05 | 4.12 | 4.16 |

Crucial rem. An automorphism of order $m$ reduces $q | J_{\text{jac}}$ by a factor $m$ and the linear algebra by $m^2$.

Timings

**Magma** 2.3 on Pentium II 450 MHz, 128 Mb.

**Example 1:**
- field $\mathbb{F}_{193^4}$
- equation $y^5 + y = z^{11}$
- genus 6
- auto 13
- $J_{\text{jac}}$ $10^{10}$
- basis 193, 445
- matrix 165, 778 × 165, 779

<table>
<thead>
<tr>
<th>construct. basis</th>
<th>1638 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>relations</td>
<td>6 days</td>
</tr>
<tr>
<td>SGE</td>
<td>1218 s</td>
</tr>
<tr>
<td>Lanczos</td>
<td>9 days</td>
</tr>
</tbody>
</table>

**Example 2:**
- field $\mathbb{F}_{193^4}$
- equation $z^2 + (z + 1)y = z^6 + 2z^4 + z^2 + 2 + z^4 + z^2 + z^2 + z + 1$
- genus 6
- auto 13
- $J_{\text{jac}}$ $10^{10}$
- basis 193, 445
- matrix 165, 778 × 165, 779

<table>
<thead>
<tr>
<th>construct. basis</th>
<th>6575 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>relations</td>
<td>9 days</td>
</tr>
<tr>
<td>SGE</td>
<td>880 s</td>
</tr>
<tr>
<td>Lanczos</td>
<td>12 days</td>
</tr>
</tbody>
</table>
Perspectives

Proven analyses: (Eng & Gaudry) unification of the methods used for finite fields and class groups.

Limiting factor: linear algebra stage, not the collection of relations (no factorization). Wait for progress in distributed linear algebra.

Ex. (Sakai: Sakurai: Smart) $y^2 + y = x^{10} + x^{11} + x^9 + x^8 + 1$ (genus 6) on $\mathbb{F}_p$, with an automorphism of order $4 \times 29$. gJJac $p^2 = 2^{17}$. Requires $gF = 4$, 500, 000.

Genus 4: rho and PG’s are $O(q^3)$; dividing the size of the basis by $g^{1/2}$ yields a $q^{1/2}$ method.

Genus 2: (Gaudry) using a union-find type method for performing the linear algebra phase, same complexity as rho.

Genus 3: linear time linear algebra?

Important application: use this as a tool in the Weil descent.

Weil descent: an example

$$E : Y^2 = X^3 + aX + \beta$$

Write $a = \sum a_i X_i$, $\beta = \sum b_i Y_i$.

$$X = \sum x_i X_i, \quad Y = \sum y_i Y_i,$$

$$0 = Y^2 - (X^3 + aX + \beta) = \sum \varphi_j (a_i, b_i, x_i, y_i) X_i$$

Thm. $\{ \psi_i \}$ defines an abelian variety of dimension $n$, with law (and points) inherited from $E$. Moreover, $\chi_A(T) = \chi_{\psi}(T^n)$.

A nice case: $2 \mid q$, $E_{\psi}$, with basis $\{ \psi, \theta, \ldots, \theta^{q-1} \}$ and $\theta + \theta^2 + \ldots + \theta^{q-1} = 1$. On $E : Y^2 + XY = X^3 + \beta$, write $X = \sum z_i \theta^x \ldots$ and find the equation of $C$ on $A$ s.t.

$$x_0 = x_1 = \cdots = x_{n-1} = x.$$ 

$$C : y^2 + x^{q-1} + 1 + \sum_{i=0}^{n-1} x^{2^i} + g(x) = 0$$

with $\deg(g) \leq 2^n$.

IV. Cryptographic implications

- Be very careful when using curves with an automorphism of large order.

- Find curves with cardinality easy to compute but without automorphisms: use random curves or maybe $X_g(Y)$ (which has only involutions).

- Low genus is 5 or maybe 4 or 3 (surely not 1)? Implies make the base field larger (bad news for smart cards).

- buy random elliptic curves!